

IN THE CLAIMS

Please amend claims 1-55 as set forth below.

The following amendments to the claims are made pursuant to the requirements of 37 C.F.R. § 1.121(c). A claim listing is provided beginning on the next page of this response.

1. (currently amended) A computer-implemented method for detecting an anomalous operation of a computer system, comprising:

- (a) monitoring transitions between and among program instrumentation points within an internal operating environment on the computer system and producing program execution trace data;
- (b) comparing the program execution trace data with data indicative of a nominal operation of the computer system; and
- (c) identifying an anomalous operation of the computer system based on the result of the comparison.

2. (currently amended) A The computer-implemented method as recited in claim 1, wherein said monitoring is performed by employing software signals obtained from instrumented code at instrumentation points in an execution path of the computer system.

3. (currently amended) A The computer-implemented method as recited in claim 1, wherein said monitoring is performed by employing software signals obtained from a hardware address bus associated with the computer system.

4. (currently amended) A The computer-implemented method as recited in claim 1, wherein said program execution trace data is employed to provide an execution profile including a list of execution paths that have executed in a specified time frame and the frequencies of executions.

5. (currently amended) A The computer-implemented method as recited in claim 1, wherein the computer system comprises a plurality of program modules in an instrumented software system.

6. (currently amended) A The computer-implemented method as recited in claim 5, wherein each program module implements a predefined functional requirement.

7. (currently amended) A The computer-implemented method as recited in claim 6, wherein each program module includes a mechanism for calling another module,

and the method further comprises the use of a statistical methodology to identify a relatively small set of cohesive program modules that represent the dynamic bindings among program modules as they execute.

8. (currently amended) A The computer-implemented method as recited in claim 7, wherein instrumentation points are employed to monitor the activity of an executing program and to indicate an epoch in the execution of the program.

9. (currently amended) A The computer-implemented method as recited in claim 8, further comprising recording, in an execution profile for the program, telemetry from the instrumentation points at each epoch.

10. (currently amended) A The computer-implemented method as recited in claim 9, wherein the execution profile comprises an n element vector (\mathbf{X}) comprising at least one entry for each program module.

11. (currently amended) A The computer-implemented method as recited in claim 10, wherein each element, x_i , of said vector contains a frequency count for the number of times that the corresponding instrumentation point m_i has executed during an era of k epochs, where $k = \sum_{i=1}^n x_i$.

12. (currently amended) A The computer-implemented method as recited in claim 11, wherein an execution profile is recorded whenever the number of epochs, k , reaches a predefined count, K , at which time the contents of the execution profile vector is set to zero.

13. (currently amended) A The computer-implemented method as recited in claim 12, wherein the recorded activity of the program during its last $L = jK$ epochs is stored in a sequence of j execution profiles, $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_j$, where the value $x_{i,j}$ represents the frequency of execution of the i^{th} program module on the j^{th} execution profile.

14. (currently amended) A The computer-implemented method as recited in claim 11, further comprising the step of reducing the dimensionality of the execution profiles from n , the number of instrumentation points whose activity is highly correlated, to a smaller set of m virtual instrumentation points whose activity is relatively uncorrelated.

15. (currently amended) A The computer-implemented method as recited in claim 14, wherein the statistical technique of principal components analysis is employed to reduce the dimensionality of the execution profiles.

16. (currently amended) A The computer-implemented method as recited in claim 14, wherein the statistical technique of principal factor analysis is employed to reduce the dimensionality of the execution profiles.

17. (currently amended) A The computer-implemented method as recited in claim 14, wherein an $n \times j$, $j > n$ data matrix $\mathbf{D} = \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_j$ is factored into m virtual orthogonal module components, where m is less than n , whereby the dimensionality is reduced from n to m .

18. (currently amended) A The computer-implemented method as recited in claim 17, wherein an eigenvalue λ_i is associated with each of the m orthogonal components.

19. (currently amended) A The computer-implemented method as recited in claim 18, wherein the eigenvalues satisfy the relation $\sum_{i=1}^n \lambda_i = n$.

20. (currently amended) A The computer-implemented method as recited in claim 17, further comprising using a predefined stopping rule in determining a number of components extracted in an orthogonal structure representing an execution profile with reduced dimensionality.

21. (currently amended) A The computer-implemented method as recited in claim 20, wherein the stopping rule is: extract all components whose eigenvalues are greater than a predefined threshold.

22. (currently amended) A The computer-implemented method as recited in claim 20, wherein the stopping rule is: extract those components such that the proportion of variation represented by $v = \frac{1}{n} \sum_{i=1}^m \lambda_i$ is at least equal to a predefined value.

23. (currently amended) A The computer-implemented method as recited in claim 17, further comprising constructing a matrix (**P**), wherein said matrix is an $n \times m$ structure whose rows, $p_{\cdot j}$, contain values showing the degree of relationship of the variation of the i^{th} program module and the j^{th} factor or principal component.

24. (currently amended) A The computer-implemented method as recited in claim 17, further comprising the step of forming a mapping vector (**O**) for at least one execution profile vector.

25. (currently amended) A The computer-implemented method as recited in claim 24, wherein the mapping vector, **O**, comprises elements o_j whose values are defined as follows:

$$\text{let } q_i = \max_{1 \leq j \leq m} p_{ij};$$

let $o_j = \text{index}(q_j)$ represent the column number in which the corresponding value q_j occurs.

26. (currently amended) A The computer-implemented method as recited in claim 25, wherein the mapping vector contains data to map probe event frequencies recorded in the execution profile vector onto corresponding virtual module equivalents.

27. (currently amended) A The computer-implemented method as recited in claim 26, wherein a frequency count for each instrumentation point k in an execution

profile vector is represented by a value f_k , and the mapping vector element o_k contains an the index value that k maps into.

28. (currently amended) A The computer-implemented method as recited in claim 17, wherein m orthogonal sources of variation in the data vector \mathbf{D} representing the original n program instrumentation points are identified.

29. (currently amended) A The computer-implemented method as recited in claim 27, wherein, on each of the original raw execution profiles, the instrumentation point frequency count is represented in the elements, $x_{i,j}$, of the profile vector, \mathbf{X}_i .

30. (currently amended) A The computer-implemented method as recited in claim 24, wherein a frequency count for each instrumentation point k in an execution profile vector is represented by a value f_k ; wherein the mapping vector element o_k contains an the index value that k maps into; wherein the mapping vector contains data to map probe event frequencies recorded in the execution profile vector onto corresponding virtual module equivalents; and wherein, after the mapping vector has been established, a virtual profile vector (\mathbf{Y}_i) is employed to contain the frequency counts for interactions among virtual execution domain sets.

31. (currently amended) A The computer-implemented method as recited in claim 30, wherein the virtual profile vector, \mathbf{Y}_i , is defined by:

$$y_{k,\bullet} = \sum_{i=1}^n f(x_{i,\bullet}) \text{ where,}$$

$$f(x_{i,\bullet}) = \begin{cases} 0 & \text{if } o_i \neq k \\ x_{i,\bullet} & \text{otherwise} \end{cases}$$

32. (currently amended) A computer-implemented method for detecting an anomalous operation of a computer system including a plurality of program modules, comprising:

- (a) monitoring transitions between and among instrumentation points within an internal operating environment on the computer system, wherein said monitoring is performed by employing software signals obtained from instrumented code in the program modules;
- (b) providing program instrumentation trace data representative of the transitions between and among program modules within a time frame;
- (c) identifying a relatively small set of virtual execution domains whose activity is substantially uncorrelated, and using this information to reduce the amount of trace data needed to detect anomalous activity;
- (d) comparing the reduced amount of trace data with predefined data indicative of a nominal operation of the computer system; and
- (e) identifying an anomalous operation of the computer system based on the result of the comparison.

33. (currently amended) A The computer-implemented method as recited in claim 32, wherein said program execution trace data is employed to provide an execution profile including a list of execution paths that have executed in a specified time frame and the frequencies of executions.

34. (currently amended) A The computer-implemented method as recited in claim 32, wherein each program module includes a mechanism for calling another module, and wherein step (c) comprises the use of a statistical methodology to identify a relatively small set of cohesive program modules that represent dynamic bindings among program modules as they execute.

35. (currently amended) A The computer-implemented method as recited in claim 32, wherein instrumentation points are employed to monitor the activity of an executing program and to indicate an epoch in the execution of the program.

36. (currently amended) A The computer-implemented method as recited in claim 35, further comprising recording, in a first execution profile for the program, telemetry from the instrumentation points at each epoch.

37. (currently amended) A The computer-implemented method as recited in claim 36, wherein the first execution profile comprises an n element vector (\mathbf{X}) comprising at least one entry for each program module, and wherein each element, x_i , of said vector contains a frequency count for the number of times that the corresponding instrumentation point m_i has executed during an era of k epochs, where $k = \sum_{i=1}^n x_i$; and wherein an execution profile is recorded whenever the number of epochs, k , reaches a predefined count, K , at which time the contents of the execution profile vector is set to zero.

38. (currently amended) A The computer-implemented method as recited in claim 37, wherein the recorded activity of the program during its last $L = jK$ epochs is stored in a sequence of j execution profiles, $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_j$, where the value $x_{i,j}$ represents the frequency of execution of the i^{th} program module on the j^{th} execution profile.

39. (currently amended) A The computer-implemented method as recited in claim 38, further comprising the step of reducing the dimensionality of the execution profiles from n , the number of instrumentation points whose activity is highly correlated, to a smaller set of m virtual instrumentation points whose activity is relatively uncorrelated.

40. (currently amended) A The computer-implemented method as recited in claim 39, wherein the statistical technique of principal components analysis is employed to reduce the dimensionality of the execution profiles.

41. (currently amended) A The computer-implemented method as recited in claim 39, wherein the statistical technique of principal factor analysis is employed to reduce the dimensionality of the execution profiles.

42. (currently amended) A The computer-implemented method as recited in claim 39, wherein an $n \times j$, $j > n$ data matrix $\mathbf{D} = \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_j$ is factored into m virtual orthogonal module components, where m is less than n , whereby the dimensionality is reduced from n to m .

43. (currently amended) A The computer-implemented method as recited in claim 42, wherein an eigenvalue λ_i is associated with each of the m orthogonal components.

44. (currently amended) A The computer-implemented method as recited in claim 43, wherein the eigenvalues satisfy the relation $\sum_{i=1}^n \lambda_i = n$.

45. (currently amended) A The computer-implemented method as recited in claim 39, further comprising using a predefined stopping rule in determining a number of components extracted in an orthogonal structure representing an execution profile with reduced dimensionality.

46. (currently amended) A The computer-implemented method as recited in claim 45, wherein the stopping rule is: extract all components whose eigenvalues are greater than a predefined threshold.

47. (currently amended) A The computer-implemented method as recited in claim 45, wherein the stopping rule is: extract those components such that the proportion of variation represented by $v = \frac{1}{n} \sum_{i=1}^m \lambda_i$ is at least equal to a predefined value.

48. (currently amended) A The computer-implemented method as recited in claim 45, further comprising constructing a matrix (**P**), wherein said matrix is an $n \times m$ structure whose rows, $p_{\cdot j}$, contain values showing the degree of relationship of the variation of the i^{th} program module and the j^{th} factor or principal component.

49. (currently amended) A The computer-implemented method as recited in claim 48, further comprising the step of forming a mapping vector (**O**) for at least one execution profile vector, wherein the mapping vector, **O**, comprises elements o_j whose values are defined as follows:

$$\text{let } q_i = \max_{1 \leq j \leq m} p_{ij};$$

let $o_j = \text{index}(q_j)$ represent the column number in which the corresponding value q_j occurs.

50. (currently amended) A The computer-implemented method as recited in claim 49, wherein the mapping vector contains data to map probe event frequencies recorded in the execution profile vector onto corresponding virtual module equivalents.

51. (currently amended) A The computer-implemented method as recited in claim 50, wherein a frequency count for each instrumentation point k in an execution profile vector is represented by a value f_k , and the mapping vector element o_k contains an the index value that k maps into.

52. (currently amended) A The computer-implemented method as recited in claim 51, wherein m orthogonal sources of variation in the data vector \mathbf{D} representing the original n program instrumentation points are identified.

53. (currently amended) A The computer-implemented method as recited in claim 52, wherein, on each of the original raw execution profiles, the instrumentation point frequency count is represented in the elements, $x_{i,j}$, of the profile vector, \mathbf{X}_i .

54. (currently amended) A The computer-implemented method as recited in claim 53, wherein the mapping vector contains data to map probe event frequencies recorded in the execution profile vector onto corresponding virtual module equivalents; and wherein, after the mapping vector has been established, a virtual profile vector (\mathbf{Y}_i) is employed to contain the frequency counts for interactions among virtual execution domain sets.

55. (currently amended) A The computer-implemented method as recited in claim 54, wherein the virtual profile vector, \mathbf{Y}_i , is defined by:

$$y_{k,\bullet} = \sum_{i=1}^n f(x_{i,\bullet}) \text{ where,}$$

$$f(x_{i,\bullet}) = \begin{cases} 0 & \text{if } o_i \neq k \\ x_{i,\bullet} & \text{otherwise} \end{cases}$$

56. (original) A computer system, comprising:

(a) a plurality of program modules;

(b) monitoring means for monitoring transitions between and among instrumentation points within the program modules, wherein said monitoring is performed by employing software signals obtained from instrumented code in the program modules, and for providing program instrumentation trace data representative of the transitions between and among program modules within a time frame;

(c) means for identifying a relatively small set of virtual execution domains whose activity is substantially uncorrelated, and using this information to reduce the amount of trace data needed to detect anomalous activity;

(d) means for comparing the reduced amount of trace data with predefined data indicative of a nominal operation of the computer system; and

(e) means for identifying an anomalous operation of the computer system based on the result of the comparison.

57. (original) A system as recited in claim 56, wherein said program execution trace data is employed to provide an execution profile including a list of execution paths that have executed in a specified time frame and the frequencies of executions.

58. (original) A system as recited in claim 56, wherein each program module includes a mechanism for calling another module, and wherein step (c) comprises the use of a statistical method to identify a relatively small set of cohesive program modules that represent dynamic bindings among program modules as they execute.

59. (original) A system as recited in claim 56, wherein instrumentation points are employed to monitor the activity of an executing program and to indicate an epoch in the execution of the program.

60. (original) A system as recited in claim 59, further comprising recording, in a first execution profile for the program, telemetry from the instrumentation points at each epoch.

61. (original) A system as recited in claim 60, wherein the first execution profile comprises an n element vector (\mathbf{X}) comprising at least one entry for each program module, and wherein each element, x_i , of said vector contains a frequency count for the number of times that the corresponding instrumentation point m_i has executed during an era of k epochs, where $k = \sum_{i=1}^n x_i$; and wherein an execution profile is recorded whenever the number of epochs, k , reaches a predefined count, K , at which time the contents of the execution profile vector is set to zero.

62. (original) A system as recited in claim 61, wherein the recorded activity of the program during its last $L = jK$ epochs is stored in a sequence of j execution profiles, $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_j$, where the value $x_{i,j}$ represents the frequency of execution of the i^{th} program module on the j^{th} execution profile.

63. (original) A system as recited in claim 62, further comprising the step of reducing the dimensionality of the execution profiles from n , the number of instrumentation points whose activity is highly correlated, to a smaller set of m virtual instrumentation points whose activity is relatively uncorrelated.

64. (original) A system as recited in claim 63, wherein the statistical technique of principal components analysis is employed to reduce the dimensionality of the execution profiles.

65. (original) A system as recited in claim 63, wherein the statistical technique of principal factor analysis is employed to reduce the dimensionality of the execution profiles.

66. (original) A system as recited in claim 63, wherein an $n \times j$, $j > n$ data matrix $\mathbf{D} = \mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_j$ is factored into m virtual orthogonal module components, where m is less than n , whereby the dimensionality is reduced from n to m .

67. (original) A system as recited in claim 66, wherein an eigenvalue λ_i is associated with each of the m orthogonal components.

68. (original) A system as recited in claim 67, wherein the eigenvalues satisfy the relation $\sum_{i=1}^n \lambda_i = n$.

69. (original) A system as recited in claim 68, further comprising using a predefined stopping rule in determining a number of components extracted in an orthogonal structure representing an execution profile with reduced dimensionality.

70. (original) A system as recited in claim 69, wherein the stopping rule is: extract all components whose eigenvalues are greater than a predefined threshold.

71. (original) A system as recited in claim 69, wherein the stopping rule is: extract those components such that the proportion of variation represented by $v = \frac{1}{n} \sum_{i=1}^m \lambda_i$ is at least equal to a predefined value.

72. (original) A system as recited in claim 68, further comprising constructing a matrix (\mathbf{P}) , wherein said matrix is an $n \times m$ structure whose rows, $p_{i,j}$, contain values showing the degree of relationship of the variation of the i^{th} program module and the j^{th} factor or principal component.

73. (original) A system as recited in claim 72, further comprising the step of forming a mapping vector (\mathbf{O}) for at least one execution profile vector, wherein the mapping vector, \mathbf{O} , comprises elements o_j whose values are defined as follows:

$$\text{let } q_i = \max_{1 \leq j \leq m} p_{ij};$$

let $o_j = \text{index}(q_j)$ represent the column number in which the corresponding value q_j occurs.

74. (original) A system as recited in claim 73, wherein the mapping vector contains data to map probe event frequencies recorded in the execution profile vector onto corresponding virtual module equivalents.

75. (original) A system as recited in claim 74, wherein a frequency count for each instrumentation point k in an execution profile vector is represented by a value f_k , and the mapping vector element o_k contains an the index value that k maps into.

76. (original) A system as recited in claim 75, wherein m orthogonal sources of variation in the data vector \mathbf{D} representing the original n program instrumentation points are identified.

77. (original) A system as recited in claim 76, wherein, on each of the original raw execution profiles, the instrumentation point frequency count is represented in the elements, $x_{i,j}$, of the profile vector, \mathbf{X}_i .

78. (original) A system as recited in claim 77, wherein the mapping vector contains data to map probe event frequencies recorded in the execution profile vector onto corresponding virtual module equivalents; and wherein, after the mapping vector has been established, a virtual profile vector (\mathbf{Y}_i) is employed to contain the frequency counts for interactions among virtual execution domain sets.

79. (original) A system as recited in claim 78, wherein the virtual profile vector, \mathbf{Y}_i , is defined by:

$$y_{k,\bullet} = \sum_{i=1}^n f(x_{i,\bullet}) \text{ where,}$$

$$f(x_{i,\bullet}) = \begin{cases} 0 & \text{if } o_i \neq k \\ x_{i,\bullet} & \text{otherwise} \end{cases}.$$